

Please replace paragraph [0034] with the following replacement paragraph:

[0034] Referring therefore to Figure 4, the process of the client 300 requesting meta-data is shown generally by the numeral 420. The client first requests at step 422 the meta-data of interest through the sandbox 306. At step 424, the client 300 checks its meta-data cache. If the data is found at step 426 then it is returned to the client 300 at step 450. If not, then at step 428, the client version manager 302 requests the data from the proxy server. At step 430, the proxy server checks the proxy cache [[304]] 202 for the data of interest. If the data is found at step 432, then the version is put in the client meta-data cache at step 448 and, passed to the client at step 450. If the version is not found, then at step 434 the proxy server requests the data from the central server. The central server first checks the central server cache for the data at step 436. If the data is found at step 438, then data proxy server updates its cache with the data at step 446, updates the client cache at step 448 and sends the data to the client at step 450. If the data is not found, then the central server requests the data from the repository 102 at step 440. The central server cache is populated with the data at step 442. The data is then placed in the proxy server cache at step 446, the client cache at step 448 and provided to the client at step 450.

Please replace paragraph [0040] with the following replacement paragraph:

[0040] Upon receipt of a request for data, the proxy cache 202a performs the steps of Figure 7, as shown generally by the numeral 500. At step 502, the proxy cache 202a receives a request for the data. The proxy cache 202a retrieves at step 504 any meta-data necessary to fulfil fulfill the request. If the request is for bulk data, the proxy cache 202a retrieves the corresponding meta-data. At step 506, the proxy cache 202a checks the meta-data to see if the client 300 has permission to access the data. If the request is not allowed at step 508, then the proxy cache denies access to the data at step 510. If the request is allowed at step [[510]] 508, then the proxy cache 202a first retrieves any bulk data necessary to fulfil fulfill the request of step 512, and provides the data at step 514.

Please replace paragraph [0042] with the following replacement paragraph:

[0042] In another embodiment, enhanced security is provided by virtue of the provision of proxy server 200. In this embodiment, the central server 100 only accepts connections from proxy servers 200. It will not accept connections from clients 300. This configuration provides enhanced security since all communication from clients 300 use proxy servers 200. In addition, the connections between proxy servers 200 and the central server 100 may then be secured, for example using [[SSL]] Secure Sockets Layer (SSL). This provides security over the wide area network while only requiring one secure connection for all of the clients 300 attached to each proxy server 200.

Amendments to the Claims

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of claims:

1. (currently amended) A version control system for managing versioned files used in software development, the version control system comprising:

a) a central server adapted to store storing a repository of said versioned files, said central server also being adapted to control all modifications to said versioned files;

b) a at-least-one proxy server connected to said central server, said -each proxy server including a read-only cache adapted to store for storing data obtained from said repository through communication with said central server; and

c) at least one client connected to each of said proxies; said proxy server, said at least one client comprising a version manager adapted to generate requests for said versioned files through said proxy server to reduce network traffic between said proxy server and said central server and to reduce load on said central server wherein modifications to said versioned files may only be made by said central server.

2. (currently amended) A version control system according to claim 1 further comprising a plurality of branches and a plurality of proxy servers;[[,]] wherein, for a first branch of said plurality of branches, said central server is adapted to store includes a list of proxy servers selected from said plurality of proxy servers and associated with said first branch proxies for each branch in; and wherein the version control system and is configured to send an update informer to notify each proxy server in the list when a change is made to said first each branch.

3. (currently amended) A version control system according to claim 1, wherein said central server includes an access control system adapted to validate requests received by said central server.

4. (currently amended) A version control system according to claim 1, wherein said at least one client is adapted to obtain a version of a file in said repository of said versioned obtains versions of files by requesting [[them]] said version from the corresponding proxy server[[,

and]]; the proxy provides server being adapted to provide the version from [[its]] said read-only cache when available; and by requesting the version from the central server otherwise.

5. (currently amended) A version control system according to claim 1, wherein said proxy server includes a mechanism for providing versions of files a desired version of a file in said repository of said versioned files to said at least one client connected clients using said read-only cache when the desired version is available from said read-only cache, and by requesting the desired version from the central server otherwise.

6. (currently amended) A version control system according to claim 1, wherein the clients at least one client is adapted to modify the repository through said central server.

7. (original) A version control system according to claim 6, wherein the central server includes a checkout mechanism for controlling modification to the repository.

8. (original) A version control system according to claim 7, wherein the central server includes a log of changes made to the repository.

9. (currently amended) A version control system[[,]] according to claim 8, wherein the log is used to update said [[a]] proxy server after a disruption to the connection between the proxy server and the central server.

10. (currently amended) A version control system according to claim 1, wherein further comprising

a plurality of chained together proxy servers adapted to serve a geographical area,
wherein each proxy server of said plurality of chained together proxy servers is
connected to at least one client, and

a regional proxy server [[is]] connected to said central server and to [[a]] each proxy
server of said plurality of chained together proxy servers server in a geographic
area, each of the plurality of proxy servers being connected to at least one client
wherein updates from the central server to the plurality of chained together proxy servers

are first sent to the regional proxy.

11. (currently amended) A method of modifying a repository of versions of files in a version control system used in software development, said version control system including a central server and [[a]] a first interconnected client interconnected to said central server through a first intervening proxy, a second client, and a second proxy, said method comprising the following steps [[of]]:

- a) the first interconnected client requesting from the central server, through the first intervening proxy, a lock on a version of a file in the version control system;
- b) the central server checking whether the requested-version is unlocked and, if the version is unlocked, and if so granting the request of said first interconnected client through said first intervening proxy; and
- c) the central server sending an update to said second client and to said second proxy other portions of the system.

12. (currently amended) A method according to claim 11, wherein said lock permits only said first interconnected client to modify prevents other clients from modifying said version of said file.

13. (currently amended) A method according to claim 11, further comprising the step of said first interconnected client performing modification to modifying said version of said file and returning the modification to said central server through said first intervening proxy.

14. (currently amended) A method according to claim 13, further comprising the step of said central server sending said modification to said second client and to said second proxy other portions of the system.

15. (currently amended) A central server in a version control system used in software development, the version control system including a proxy servers connected to clients server having a read-only cache and connecting a client to the central server, the central server comprising the following:

- a) a repository of versioned files used in said software development;
- b) a version manager adapted to provide a version of a file for providing versions of files from said repository to said proxy server;
- c) an access control system for managing requests adapted to manage a request from the client clients to modify the repository; and
- d) a memory structure adapted to store a log of modifications changes made to the repository[[;]] and adapted to store [[e]] a list of connected proxies and portions of said repository stored in the read-only cache of the proxy server, the proxies containing read-only caches of said portions of said repository for providing versions of files to said clients.

16. (currently amended) A central server according to claim 15, wherein said log is used to update one of said read-only cache eacheis in said a-respective proxy server after a disruption to the connection between said respective proxy server and said central server.

17. (currently amended) A proxy server in a version control system used in software development, the version control system including a client and a central server containing a repository of versioned files and a client, said proxy server comprising the following:

- a) a read-only cache adapted to store a requested version of a file requested by the client and received for storing data from said repository; and
- b) a version provider adapted to provide [[a]] the requested version of [[a]] the file to said client, the version provider being configured to first check the read-only cache for the requested version and, if [[it]] the requested version is not found, to request the requested version from said central server on behalf of said client.

18. (currently amended) A proxy server according to claim 17, wherein the read-only cache is configured to store copies of version a plurality of versions of files requested from said central server by said client.

19. (currently amended) A computer-readable medium containing processor instructions for implementing a version control system to be used in software development, including instructions for the following steps:

- a) a central server storing a repository of versioned files on a central server configured to control all modifications to said versioned files;
- b) establishing communication between at least one proxy connected to said central server and a each proxy server including a read-only cache configured to store for storing data obtained from said repository through communication between said central server and said proxy server; and
- c) establishing communication between said proxy server and a at-least-one client connected to each of said proxies; comprising a version manager adapted to generate a request for a versioned file from said repository through said proxy server to reduce network traffic between said central server and said client and to reduce load on said central server wherein modifications to said versioned files may only be made by said central server.

20. (currently amended) A computer-readable medium according to claim 19, further comprising instructions for the following steps:

maintaining to maintain a list of additional proxy servers proxies for each branch in the version control system, said additional proxy servers being in addition to said proxy server in communication with said client; and

notify each notifying said additional proxy servers in the list when a change is made to said versioned file each branch.

21. (currently amended) A computer-readable medium according to claim 19, wherein said central server includes an access control system adapted to validate requests received by said central server from said proxy server, said computer-readable medium further comprising instructions for the following step: validating said request received by said central server.

22. (currently amended) A computer-readable medium according to claim 19, further comprising instructions for the following steps:

commanding , wherein said version manager of said client obtains versions of files by requesting to request the versioned file them from the corresponding proxy server;[[, and]]

commanding the proxy server to provide the versioned file provides the version from [[its]] said read-only cache when available and by requesting the version from the central server

otherwise the versioned file is stored in said read-only cache, and
commanding the proxy server to request the versioned file from the central server when
the versioned file is not stored in said read-only cache.

23. (currently amended) A computer-readable medium according to claim 19, further comprising instructions for the following steps:

, wherein said proxy server includes a mechanism for providing the versioned file versions of files to the client connected clients using said read-only cache when the versioned file desired version is stored in said read-only cache available; and

said proxy server requesting the versioned file version from the central server when the versioned file is not stored in said read-only cache otherwise.

24. (currently amended) A computer-readable medium according to claim 19, further comprising instructions for the following steps:

said client requesting via said proxy server modifications to said versioned file in said repository; and

said proxy server requesting said central server to make said modification to said versioned file in said repository, wherein the clients modify the repository through said central server.

25. (currently amended) A computer-readable medium according to claim 24, further comprising instructions for the following step:

, wherein the central server includes a checkout mechanism for controlling modification [[to]] of the repository via a checkout mechanism comprising part of said central server.

26. (currently amended) A computer-readable medium according to claim 25, further comprising instructions for the following step:

keeping in wherein the central server includes a log of changes made to the repository.

27. (currently amended) A computer-readable medium[[,]] according to claim 26, further comprising instructions for the following step:

using ,wherein the log is used to update the [[a]] proxy server after a disruption to the communication connection between the proxy server and the central server.

28. (currently amended) A computer-readable medium according to claim 19, further comprising instructions for the following steps:

chaining together a plurality of proxy servers configured to serve a geographic area;
establishing communication between the plurality of chained together proxy servers and a regional proxy server;

establishing communication between said , wherein a regional proxy server and is connected to said central server and to a plurality of proxy server in a geographic area, each of the plurality of proxy servers being connected to at least one client wherein; and

sending updates from the central server [[to]] for the plurality of chained together proxy servers are first sent to the regional proxy server.

29. (new) A version control system according to claim 1, wherein said proxy server is transparent to users of said at least one client.